Description

SYSTEMS AND METHODS FOR PROVIDING PROXY CONTROL FUNCTIONS IN A WORK MACHINE

Cross Reference to Related Applications

[01] This application claims the benefit of U.S. Provisional Application Serial No. 60/483,915 entitled "Systems and Methods for Interfacing Off-Board and On-Board Networks in a Work Machine," filed July 2, 2003, owned by the assignee of this application and expressly incorporated herein by reference in its entirety.

[02] This application is related to U.S. Application No. ________, entitled "SYSTEMS AND METHODS FOR PROVIDING SERVER OPERATIONS IN A WORK MACHINE," filed August 25, 2003, U.S. Application No. ________, entitled "SYSTEMS AND METHODS FOR PROVIDING SECURITY OPERATIONS IN A WORK MACHINE," filed August 25, 2003, U.S. Application No. ________, entitled "SYSTEMS AND METHODS FOR PROVIDING NETWORK COMMUNICATIONS BETWEEN WORK MACHINES," filed August 25, 2003, and U.S. Application No. ________, entitled "SYSTEMS AND METHODS FOR PERFORMING PROTOCOL CONVERSIONS IN A WORK MACHINE," filed August 25, 2003, each owned by the assignee of this application and expressly incorporated herein by reference in its entirety.

Technical Field

[03] This invention relates generally to network interface systems and more particularly, to systems and methods for providing a virtual control module in a work machine environment.

Background

[04] An important feature in modern work machines (e.g., fixed and mobile commercial machines, such as construction machines, fixed engine systems, marine-based machines, etc.) is the on-board electronic communications, monitoring, and control network. An on-board network includes many different modules connected to different types of communication links. These links may be proprietary and non-proprietary, such as manufacturer-based data links and communication paths based on known industry standards (e.g., J1939, RS-232, RP1210, RS-422, RS-485, MODBUS, CAN, etc.). Other features associated with work machine environments are off-board networks, such as wireless networks (e.g., cellular), satellite networks (e.g., GPS), and TCP/IP-based networks.

[05] On-board modules may communicate with other on-board or off-board modules to perform various functions related to the operation of the work machine. For example, display modules may receive sensor data from an engine control module via a J1939 data link, while another control module connected to a proprietary data link may provide data to another module connected to the same link. Also, an on-board module may send data to an off-board system using a different communication path extending from the work machine to the off-board system.

[06] Existing work machine environments typically include several control modules, each having individual addresses on one or more data links. Messages intended for these modules may be broadcasted on the data link for receipt by the respective control modules. While the logical functionality of such control modules are important, the physical devices themselves and their respective locations on the network are, in many instances, irrelevant. In fact, implementing various module functionality by way of discrete physical devices may frustrate system performance, operation, and design. Implementing several discrete modules may pose problems for system maintenance and diagnostics.

Further, implementing, maintaining, and integrating several discrete modules in a work machine environment may be costly and time consuming. For example, implementing several specialized components may necessitate multiple networks and communications adapters.

[07] Moreover, problems arise when modules connected to different types of data links need to communicate. These problems become especially acute as the number of data links and protocols on a given work machine increases. For example, various protocols (e.g., J1939, RS-232, RP1210, RS-422, RS-485, MODBUS, CAN, ISO11783, ATA, etc.) may be required to accommodate specialized modules in current work machine environments. Further, problems arise when legacy systems need to communicate with other, perhaps newer, systems that are not compatible with the legacy protocols. To address these problems, conventional systems may incorporate various interface devices to facilitate communications between different types of data links. Although this solution may be functionally acceptable in some instances, their implementations are restricted due to the hardware and service capabilities associated with the types of data links used in a work machine. Further, the additional hardware may take up valuable space needed for other components used by the machine.

[08] U.S. Patent No. 6,512,970 to Albert describes an electronic control unit for use with autonomously controlled assemblies in motor vehicles. Albert's control unit uses non-overlapping memory zones, each assigned a diagnostic and programming address, and a central control device that interacts with the memory zones using time division multiplexing. Each memory zone, in conjunction with the central control device, forms a virtual control unit for a vehicle assembly. Although Albert provides a solution for providing virtual control units, the system has limited flexibility in terms of both architecture and application. The electronic control unit described by Albert uses non-overlapping memory zones and does not share code through its memory. Moreover, to provide virtual

control, the system requires time-division multiplex communication between a central control device and the non-overlapping memories. In addition, Albert's system cannot provide effective virtual control services in a multi-protocol environment in which devices require both simultaneous and discrete interaction.

[09] Methods, systems, and articles of manufacture consistent with certain embodiments of the present invention are directed to solving one or more of the problems set forth above.

Summary of the Invention

[10] Methods and systems of the present invention may be provided for performing proxy control functions in an environment including one or more work machines, each including one or more modules coupled to one or more data links. Consistent with embodiments of the present invention, a gateway, located in a work machine, may monitor a data link (e.g., a proprietary data link) for broadcasted messages. The messages may include source and/or destination address identifiers. In exemplary embodiments of the present invention, the gateway may intercept a message from the data link based on the source and/or the destination address included in the message. Upon receiving a message from the data link, the gateway may use a mapping structure to route the message to proxy logic that performs functions associated with the destination module. The proxy logic may be located in the gateway and may perform functions that are similar to the destination module.

[11] Consistent with certain embodiments of the present invention, methods and system may receive a message from a first data link (e.g., an RS-232 data link) interfaced by control logic included in a gateway. The message may be intended for a destination module. Methods and systems may identify, via an address map, the location of the destination module in a work machine environment. The message may be formatted and routed, via the address map, to the destination module over a second data link (e.g., a proprietary data link) coupled to the gateway.

Brief Description of the Drawings

[12] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several aspects of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[13] Fig. 1 is a block diagram of an exemplary system that may be configured to perform certain functions consistent with embodiments of the present invention;

[14] Fig. 2 is a block diagram of an exemplary gateway consistent with embodiments of the present invention;

[15] Fig. 3 is a block diagram of an exemplary software architecture for a gateway consistent with embodiments of the present invention;

[16] Fig. 4 is a block diagram of an exemplary work machine environment consistent with embodiments of the present invention;

[17] Fig. 5 is a block diagram of an exemplary gateway that is configured as a proxy control module consistent with embodiments of the present invention;

[18] Fig. 6 is a flowchart of an exemplary proxy control process consistent with embodiments of the present invention;

[19] Fig. 7 is a block diagram of an exemplary work machine environment consistent with embodiments of the present invention;

[20] Fig. 8 is a flowchart of an exemplary process consistent with embodiments of the present invention; and

[21] Fig. 9 is a block diagram of an exemplary controller-based work machine environment consistent with embodiments of the present invention.

Detailed Description

[22] Reference will now be made in detail to exemplary aspects of the present invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Overview

[23] Fig. 1 illustrates an exemplary system 100 in which features and principles consistent with an embodiment of the present invention may be implemented. As shown in Fig. 1, system 100 may include a work machine 105 including an on-board system 110 comprising a gateway 120 and on-board modules 125, 127. System 100 may also include one or more off-board systems 130-150. Although gateway 120 is shown as a separate element, methods and systems consistent with the present invention may allow gateway 120 to be included in one or more elements, such as on-board modules 125 and/or 127.

[24] A work machine, as used herein, refers to a fixed or mobile machine that performs some type of operation associated with a particular industry, such as mining, construction, farming, etc. and operates between or within work environments (e.g., construction site, mine site, power plant, etc.). A non-limiting example of a fixed machine includes an engine system operating in a plant, off-shore environment (e.g., off-shore drilling platform). Non-limiting examples of mobile machines include commercial machines, such as trucks, cranes, earth moving vehicles, mining vehicles, backhoes, material handling equipment, farming equipment, marine vessels, aircraft, and any type of movable machine that operates in a work environment.

[25] An on-board module, as used herein, may represent any type of component operating in work machine 105 that controls or is controlled by other components or sub-components. For example, an on-board module may be an operator display device, an Engine Control Module (ECM), a power system

control module, a Global Positioning System (GPS) interface device, an attachment interface that connects one or more sub-components, and any other type of device work machine 105 may use to facilitate operations of the machine during run time or non-run time conditions (i.e., machine engine running or not running, respectively).

[26] An off-board system, as used herein, may represent a system that is located remote from work machine 105. An off-board system may be a system that connects to on-board system 110 through wireline or wireless data links. Further, an off-board system may be a computer system including known computing components, such as one or more processors, software, display, and interface devices that operate collectively to perform one or more processes. Alternatively, or additionally, an off-board system may include one or more communications devices that facilitates the transmission of data to and from on-board system 110.

[27] Gateway 120 represents one or more interface devices configured to perform functions consistent with various embodiments of the present invention. Gateway 120 may be configured with various types of hardware and software depending on its application within a work machine. Thus, in accordance with embodiments of the invention, gateway 120 may provide interface capability that facilitates the transmission of data to and from on-board system 110, performs various data processing functions, and maintains data for use by one or more on-board modules or off-board systems. For example, gateway 120 may be configured to perform protocol conversions (e.g., tunneling and translations), intelligent routing, and server-based operations, such as data provisioning, application provisioning, Web server operations, electronic mail server operations, data traffic management, and any other type of server-based operations that enable on-board system 110 to retrieve, generate, and/or provide data with on-board and/or off-board systems. For clarity of explanation, Fig. 1 depicts gateway 120 as a distinct element. However, consistent with principles of

the present invention, "gateway" functionality may be implemented via software, hardware, and/or firmware within one or more modules (e.g., 125 and/or 127) on a network, which controls a system on a work machine and communicates with an off-board system. Thus, gateway 120 may, in certain embodiments, represent functionality or logic embedded within another element.

[28] On-board module 125 represents one or more on-board modules connected to one or more proprietary data links 128 included in on-board system 110. On-board module 127 may be one or more on-board modules connected to a non-proprietary data link 129, such as Society of Automotive Engineers (SAE) standard data links including Controller Area Network (CAN), J1939, etc. standard data links.

[29] As shown in Fig. 1, gateway 120 also interfaces with one or more off-board systems 130-150. In one exemplary embodiment, off-board systems 130-150 include, for example, computer system 130, computer system 140, and service port system 150.

[30] Computer system 130 represents one or more computing systems each executing one or more software applications. For example, computer system 130 may be a workstation, personal digital assistant, laptop, mainframe, etc. Computer system 130 may include Web browser software that requests and receives data from a server when executed by a processor and displays content to a user operating the system. In one embodiment of the invention, computer system 130 is connected to on-board system 110 through one or more wireline based data links, such as a Local Area Network (LAN), an Extranet, and the Internet using an Ethernet connection based on TCP/IP.

[31] Computer system 140 also represents one or more computing systems each executing one or more software applications. Computer system 140 may be a workstation, personal digital assistant, laptop, mainframe, etc. Also, computer system 140 may include Web browser software that requests and receives data from a server when executed by a processor and displays content to

a user operating the system. In one embodiment of the invention, computer system 140 is connected to on-board system 110 through one or more wireless based data links, such as cellular, satellite, and radio-based communication data links.

[32] Computer systems 130 and 140 may each be associated with a user (e.g., customer), multiple users, a business entity (dealer, manufacturer, vendor, etc.), a department of a business entity (e.g., service center, operations support center, logistics center, etc.), and any other type of entity that sends and/or receives information to/from on-board system 110. Further, computer system 130 and 140 may each execute off-board software applications that download or upload information to/from on-board system 110 via gateway 120. In certain embodiments, computer systems 130 and 140 may include one or more controllers such as a PLC (Programmable Logic Controller), which could be used in plants/factories.

[33] Service system 150 represent one or more portable, or fixed, service systems that perform diagnostics and/or service operations that include receiving and sending messages to on-board system 110 via gateway 120. For example, service system 150 may be a electronic testing device that connects to on-board system 120 through an RS-232 serial data link. Using service system 150, a user or an application executed by a processor may perform diagnostics and service operations on any of on-board system modules 125, 127 through gateway 120.

[34] In one embodiment, gateway 120 may include various computing components used to perform server based services (e.g., communications services, file services, database services, etc.) for on-board system 110. Fig. 2 shows an exemplary block diagram of gateway 120 consistent with embodiments of the present invention. As shown, gateway 120 includes a digital core 202, on-board data link port components 220-1 to 220-N, and off-board data link port components 225-1 to 225-Y.

[35] Digital core 202 includes the logic and processing components used by gateway 120 to perform its interface, communications, and server functionalities. In one embodiment, digital core 202 includes one or more processors 205 and internal memories 210 and 215. Processor 205 may represent one or more microprocessors that execute software to perform the gateway features of the present invention. Memory 210 may represent one or more memory devices that temporarily store data, instructions, and executable code, or any combination thereof, used by processor 205. Memory 215 may represent one or more memory devices that store data temporarily during operation of gateway 120, such as a cache memory, register devices, buffers, queuing memory devices, and any type of memory device that maintains information. Memories 210 and 215 may be any type of memory device, such as flash memory, Static Random Access Memory (SRAM), and battery backed non-volatile memory devices.

[36] On-board data link ports 220-1 to 220-N represent one or more interface devices that interconnect one or more on-board data links with digital core 202. For example, on-board data link ports 220-1 to 220-N may connect to proprietary and non-proprietary data links 128, 129, respectively. In one embodiment, on-board data link ports 220-1 to 220-N interfaces with one or more proprietary data links, one or more CAN data links (e.g., J1939, galvanized isolated CAN data links, etc.), one or more RS-232 serial based data links (e.g., MODBUS, PPP, NMEA183, etc.), and one or more RS-242 data links. On-board data link ports 220-1 to 220-N may also include virtual (i.e., software) ports that allow a single connection to act as if there were multiple connections.

[37] Off-board data link ports 225-1 to 225-Y represent one or more interface devices that interconnect one or more off-board data links with digital core 202. For example, off-board data link ports 225-1 to 225-Y may connect gateway 120 to one or more RS-232 data links, RS-485 data links, Ethernet data links, MODBUS data links, radio data links, and/or satellite data links, etc. It is

appreciated that gateway 120 may be configured to interface with any type of data link used in an on-board or off-board system network.

[38] The gateway 120 shown in Fig. 2 is exemplary and not intended to be limiting. A number of additional components may be included in gateway 120 that supplement and/or compliment the operations of digital core 202 and data link ports 220 and 225. For example, gateway 120 may also include an internal power supply, a real time clock, hour meter, sensor inputs for receiving signals from one or more sensors monitoring the operations of a work machine component, memory arrays, etc. Moreover, as mentioned above, gateway 120 may, in certain embodiments, be implemented (e.g., via logic and/or circuitry) within one or more modules coupled to a given network.

[39] In operation, digital core 202 executes program code to facilitate communications between on-board modules and/or off-board systems. In one embodiment of the present invention, memory 210 includes application and server-based software programs that allow information received through either data link ports 220 and 225 to be processed and/or transferred to the proper destination module/system in the proper format. Fig. 3 illustrates an exemplary software architecture model 300 that may be implemented by gateway 120 consistent with embodiments of the present invention.

[40] Exemplary model 300 may include hardware interface software, such as boot executable software and driver software layer 310, that drive the on-board and off-board data link ports 220 and 225 connecting the multiple types of data links to gateway 120 (e.g., Ethernet, RS-232, CAN, proprietary data links, etc.). A core hardware access layer 315 interfaces boot executable layer 310 and core software layer 330, which includes software associated with runtime operations of gateway 120. Layer 320 includes operating system software executed by processor 205, and layer 325 is a network stack level including one or more protocol stacks used to perform communication services, such as formatting data messages for specific protocols, etc. In one embodiment, model

300 may also include a Web server layer 335 that includes server software used by gateway 120 to perform Web server operations, such as HTML processing, content generation, Web page request processing, etc. Further, model 300 may also include one or more layers 340-360 representing application programs executable by gateway 120. For example, layers 340, 345 may represent server applications executed by gateway 120 to perform certain services, such as data provisioning, application management, traffic management, etc. Layers 360-1 to 360-X may represent application programs that perform operations associated with functions typically performed by certain types of on-board modules connected to an on-board network, such as a Customer Communication Module (CCM), a communication adapter, a GPS Interface Module (GPSIM), a third party interface software, an Engine Vision Interface Module (EVIM), and a product link module.

[41] Model 300 may also include an inter-data link gateway layer 350 that includes one or more gateway applications 350-1 to 350-T, that perform protocol conversion operations for converting information associated with one type of data link to another. The conversion operations may include protocol translation and tunneling features. Processor 205 may execute a selected one of application programs 350-1 to 350-T based on the type of format required by an outgoing data link. For example, application layer 350-1 may represent a protocol conversion program that allows data messages received in a proprietary data link to be converted to a J1939 format for transmission across a J1939 data link. Other types of conversion applications may be configured in model 300 including application layers that combine one or more protocol conversion capabilities.

Proxy Control Functions

[42] Consistent with embodiments the present invention, methods and systems may leverage one or more gateways 120 in order to provide a virtual control module in a work machine network. As used herein, the term "virtual

control module" refers to a single module (e.g., a hardware device) that virtually represents multiple control modules on a data link. In such embodiments, gateway 120 may include hardware, firmware, and/or software (e.g., application programs 360-1 to 360-X) for performing operations associated with functions performed by various on-board modules connected to an on-board network. For example, gateway 120 may be programmed with logic corresponding to a CCM, a communication adapter, a GPSIM, a third party interface software, an EVIM, and/or a product link module, i.e., proxy control logic. A single gateway may therefore replace a plurality of distinct control modules in a given work machine environment. With such proxy logic, gateway 120 may serve as a virtual control module in a work machine network. That is, gateway 120 may serve as a proxy device for a plurality of distinct control modules. Consistent with embodiments of the present invention, methods and systems may enable one or more gateways to intercept messages transmitted on a data link and process such messaged using appropriate proxy logic. Accordingly, a message directed to an address that would otherwise be recognized by an on-board network as associated with an actual control module will be intercepted by gateway 120 and processed accordingly. In this fashion, gateway 120 may serve as a virtual control module.

[43] Fig. 4 is a block diagram of a work machine environment 400, associated with a work machine 405 (either fixed or mobile), in which one or more gateways are leveraged to perform proxy control functions. As illustrated in Fig. 4, environment 400 may include an engine system 414 operating in work machine 405 and connected to an on-board data link 422. In certain embodiments, data link 422 may be a proprietary data link ("PDL"). Data link 422 may, however, include any type of proprietary and/or non-proprietary on-board data link, such as J1939, MODBUS, CAN, etc. Data link 422 may be similar to data link 128 and/or 129 described in connection with Fig. 1. In embodiments of the present invention, one or more on-board modules (e.g., 415, 416) may be coupled to data link 422. Modules 415 and 416 may include any

type of on-board module, component, or sub-component operating within work machine 405. For example, modules 415 and 416 may be ECMs, J1939 display devices (e.g., sensor gauges, etc.), EVIMs, on-board diagnostic systems, etc. The illustrated modules could also represent components and systems associated with work machine 405 such as propulsion, communications, and navigation systems.

[44] As illustrated in Fig. 4, work machine 405 may include a gateway 420, which may be similar in configuration and operation as gateway 120 described above in connection with Figs. 1 and 2. In system 400, gateway 420 may be coupled to data link 422 and configured as a virtual control module. As such, gateway 420 may intercept messages (from data link 422) originating from various components (e.g., module 415) and destined for control modules for which gateway 420 serves as a proxy. Gateway 420 may also receive messages originating from on-board and/or off-board components and route such messages to modules coupled to data link 422. Accordingly, gateway 420 may include an address map, software logic corresponding to one or more replaced control modules, and corresponding address identifiers for the software logic. Fig. 5 illustrates such elements which may be included in gateway 420.

[45] As illustrated in Fig. 5, gateway 420 may include a plurality of control proxy logic elements (e.g., 530, 540, 550) corresponding to the control modules for which gateway 420 is assuming functionality. For example, gateway 420 may include proxy logic corresponding to a CCM, an EVIM, and a GPSIM. Although CCM, EVIM, and GPSIM logic elements are shown in Fig. 5, gateway 420 may be programmed with different numbers and types of logic elements. In certain embodiments, proxy logic elements may be implemented via one or more memory components (e.g., memory 210). In one example, proxy logic elements may be implemented via software modules in a single software package or throughout a plurality of software packages. Each of the proxy logic elements may share information and/or code present in the software modules and/or a given software package(s). For example, a software package may include a

plurality of logic element modules and include code that is accessible to and/or shared by the modules.

[46] Consistent with embodiments of the present invention, each of the control logic elements may interface with one or more data links (e.g., 534, 544, 554) compatible with the type of operations used by the replaced control module in normal on-board system operations. Such interfaced data links may be any type of proprietary and/or non-proprietary data links and may be coupled to corresponding on-board and/or off-board modules. For instance, a CCM control module logic element may interface with an M5X or RS-232 data link for sending information to a destination module on data link 422. GPSIM logic, on the other hand, may receive (e.g., via an RS-232 data link) a message from an off-board component (e.g., a satellite device) destined for a module attached to data link 422 (e.g., 416). Each logic element included in gateway 420 may be associated with a specific identifier (e.g., 532, 542, 552). Such identifiers may include any textual, numerical, and/or symbolic element. In certain embodiments, gateway 420 may be pre-configured with logic elements according to a particular work machine environment. In addition, or as an alternative, gateway 420 may dynamically receive or retrieve logic as such logic is needed. For example, gateway 420 could be updated with additional logic at any time and may, in certain implementations, receive logic from one or more remote locations. In one embodiment, gateway 420 may receive proxy logic from a remotely located service center, further details of which are described below in connection with Fig. 9. In one instance, gateway 420 may be automatically or manually updated with additional logic when additional components and modules are added to environment 400 or when the application or purpose of work machine 405 changes.

[47] As depicted in Fig. 5, gateway 420 may include an address mapping structure 510, which may be software-implemented, that maps addresses included in incoming messages to appropriate software logic within gateway 420.

Gateway 420 may access mapping structure 510 in order to perform virtual control module operations. In certain embodiments, mapping structure 510 may be stored in a memory device within gateway 420 and accessed using one or more processing devices, such as a memory controller or a CPU device. Alternatively, mapping structure 510 may reside external to gateway 420. Mapping structure 510 may include a listing of identifiers associated with proxy logic elements included in gateway 420. Mapping structure 510 may also indicate the data link protocols interfaced by the proxy logic elements. In one exemplary implementation, mapping structure 510 may include, in addition or as an alternative to the above features, information reflective of the functionality or processing capabilities of each proxy logic element included in the gateway. In certain embodiments, mapping structure may be pre-configured according to a particular work machine environment. In addition, or as an alternative, mapping structure 510 may dynamically update its listings in response to gateway 420 receiving additional logic. That is, when gateway 420 is programmed with a given logic element, mapping structure may dynamically update its listing to include an address identifier associated with that logic element. In certain embodiments, mapping structure 510 may be used by gateway 420 to direct received messages to components coupled to data link 422. In such embodiments, mapping structure 510 may include a listing of identifiers reflecting addresses or locations of components coupled to data link 422, e.g., modules 415 and 416. Mapping structure 510 may also reflect associations between modules located with a given work machine (e.g., 405) and logic elements included in a gateway (e.g., 420). Mapping structure 510 may map components coupled to data link 422 to corresponding logic elements in gateway 420.

[48] Fig. 6 is a flowchart depicting an exemplary proxy control process consistent with embodiments of the present invention. The illustrated process may begin when a source component coupled to data link 422 broadcasts a

message over data link 422 for receipt by one or more destination modules (Step 610). For example, component 415 may transmit a message addressed to a CCM over data link 422. A "message," as used herein, may include any type of data or information, such as numeric, textual, voice, alpha-numeric, etc., values. Further, "messages" may include packaged data or files that include data, executable code, etc., and any other form of information that may be included in an electronic transmission package, signal, message, etc. Messages may also include one or more address identifiers associated with its source and/or destination. An "address identifier" may include any textual, numerical, symbolic, and/or voice recognition element identifying a source or destination for the message in which it is included. In this example (illustrated by message 511 and flow line 513), the message may include a destination address identifier of "0014."

[49] Consistent with embodiments of the present invention, gateway 420 may be configured to monitor an on-board data link for broadcasted messages. Accordingly, gateway 420 may intercept message 511 from data link 422 (Step 620). In certain embodiments, gateway 420 may be configured to intercept messages based on their source. In one example, gateway 420 may be configured to intercept messages originating from a particular module by examining a source address identifier in the broadcasted message. Upon receiving a broadcasted message, gateway 420 may extract a destination address identifier from the message (Step 630). Gateway 420 may compare or match the extracted destination address identifier with identifiers included in address map 510 (Step 640). As explained above, mapping structure 510 may include a list of address identifiers corresponding to proxy logic included in gateway 420. If the address identifier included in the message is not found in mapping structure 510 (Step 645-No), an error message may be generated and transmitted to data link 422 (Step 655). Such an error message may indicate that the message cannot be processed as addressed. In one embodiment, gateway 420 may retrieve or be

updated with additional logic as a result of a received address identifier not being found in map 510. If the message address identifier matches an identifier in map 510 (Step 645-Yes), gateway 420 may direct the message to the corresponding logic for processing (Step 650). For example, gateway 420 may direct message 511 to CCM logic 530 included in gateway 420 based on the address identifier "0014."

[50] As explained above, gateway 420 may intercept a message from data link 422 based on its source and then determine whether the message can be processed by proxy logic. In alternative embodiments, gateway 420 may retrieve messages from the data link based on a determination that the message is destined for a module for which the gateway serves as a proxy. In certain embodiments, gateway 420 may be configured to monitor the destination address portion of messages broadcasted by modules over data link 422 in order to determine whether or not a given message should be intercepted for proxy processing. Thus, the determination with respect to whether a given message can be processed by proxy logic may occur prior to the retrieval of the entire message from the data link. In this fashion, gateway 420 may intercept a message (Step 620), match the destination address (Step 640), and route the message for processing (Step 650) only upon identifying a broadcasted message destined for a module for which gateway 420 serves as a proxy.

[51] Once a message is routed to the appropriate logic in gateway 420, that logic may process the message accordingly. In certain embodiments, the proxy logic may then route the received message (or a new message responsive to the received message) over its respective interfaced data-link for receipt by another on-board or off-board component, module, or system. Messages and responses may also be sent from one proxy logic element to another proxy logic element. In addition, or as an alternative, a responsive message may be transmitted by gateway 420 from the proxy logic back to the source module (e.g., module 415).

[52] Although the process of Fig. 6 refers to receiving a message from data link 422, gateway 420 may, additionally or alternatively, receive messages destined for components coupled to data link 422. For example, referring back to Fig. 5, gateway 420 may receive, via data link 554, a message 512 originating from an off-board module (not shown) and destined for component 416 coupled to data link 422. Message 512 may be formatted according to data link 554's protocol. In one example, data link 554 may be a RS-232 data link and message 512 may be in a format consistent with the RS-232 protocol. Message 512 may also include an address identifier. For example, message 512 may include a destination address "0060," which corresponds to GPSIM logic 550. In this example (illustrated by flowline 514), gateway 420 may use mapping structure 510 to identify the appropriate destination module (e.g., 416) and direct the message to the destination module. Mapping structure 510 may, for example, indicate that GPSIM logic 550 is associated with module 416.

[53] In certain embodiments, gateway 420 may perform protocol translation processed to facilitate data exchange between incompatible data links. Gateway may be pre-configured to perform specific translations for specific data links, modules, and proxy logic, and/or may dynamically detect inconsistencies between various device and data link protocols and perform translations accordingly. Additional details of protocol translation are discussed below.

[54] As explained above, gateway 420 may provide proxy services to modules coupled to a data link located within a given work machine. Consistent with embodiments of the present invention, one or more gateways may also be leveraged to provide proxy control in an off-board work machine environment. In such an environment, a single gateway within a given work machine may process messages originating from one or more off-board modules via proxy control logic. Fig. 7 is a block diagram of an exemplary off-board environment consistent with such embodiments.

[55] As shown in Fig. 7, a work machine 710 may include a gateway 715, which may be configured, and operates, similarly to gateway 420 described in connection with Fig. 4. Work machine 710, which may be mobile or fixed, may also include one or more modules 711, which may be similar to modules 415 and 416. As illustrated, work machine 710 may include one or more on-board data links 712, which may be similar to on-board data link 422 in Fig. 4. Gateway 715 may execute one or more server applications that allow work machine 710 to communicate with one or more off-board elements, such as one or more other work machines 720, one or more external systems 730, and/or one or more external networks such as a Wide Area Satellite Wireless Network (WASWN) 740, a Wireless Local Area Network (WLAN) 750, a Wide-Area Terrestrial Wireless Network (WATWLN) 760, and/or a Wide Area Network (WAN) 770.

[56] WASWN 740 may be a satellite radio network that includes infrastructure allowing communications between one or more satellite devices and a remote system, such as computer system 140 described in connection with Fig. 1. WLAN 750 may be a wireless radio network including infrastructure that facilitates communications between one or more wireless radio devices and a remote system, such as computer system 140. WATWLN 760 may be a wireless network that includes infrastructure allowing communications between one or more cellular devices and a remote system (e.g., computer system 140). WAN 770 may be a network including the infrastructure that allows for Internet access, such as the World Wide Web.

[57] As illustrated, work machine 720 may include a gateway 725 that may be configured and operates similar to gateway 120. Work machine 720 may also include one or more modules 721, which may be similar to modules 415 and 416. Work machine 720 may be a mobile or fixed work machine connected to work machine 710 through a wireline or wireless data link 722. External system 730 may represent a remote system that communicates with gateway 715 through

a wireless or wireline data link 732, such as computer system 130, computer system 140, or service port system 150.

[58] Although Fig. 7 shows work machine 720 and external system 730 connected to work machine 710 through dedicated data links, these elements may also be configured to communicate with gateway 715 through one or more of networks 740, 750, 760, and 770.

[59] Fig. 8 is a flowchart depicting an exemplary proxy control method consistent with embodiments of the present invention. The illustrated process may begin when gateway 715 retrieves or receives a message from an external source (Step 810). For example, gateway 715 may receive a message from external system 730 via off-board data link 732. The received message may include a corresponding source and/or destination address identifier. In certain embodiments, gateway 715 may be configured to determine whether the received message should be processed locally or should be routed to another work machine for subsequent processing or additional routing. For example, gateway 715 may examine the destination address to determine whether the message is destined for a module in work machine 710 or if it can be processed by proxy logic include in gateway 715. This determination may, in one example, be facilitated by mapping structure 510. If the received message is not destined for local processing (Step 820-No), gateway 715 may forward or re-route the message (e.g., by consulting map 510) to another work machine or external system (Step 830). Additionally, or alternatively, gateway 715 may transmit an error message back to the message source, indicating that the message cannot be processed as addressed.

[60] If, however, gateway 715 determines that the received message can be locally processed (Step 820-Yes), then gateway 715 may extract the destination address identifier (Step 840). Gateway 715 may find the extracted destination address in mapping structure 510 (Step 850) and route the message to the appropriate control logic (Step 860). For example, gateway 715 may

determine that the received message is destined for an EVIM module and may, therefore, route the message to EVIM proxy logic 540. In this fashion, the source module may be unaware of the fact that gateway 715 is acting as a proxy for the destination module. That is, external systems may transmit a message destined to a particular destination module within a work machine, and gateway 715 may (transparent to the source) intercept the message and process it via appropriate proxy logic. Upon receiving the message, the proxy control logic may generate a response to the received message, and the response may be routed back to the source module via gateway 715.

[61] In addition to providing proxy services for specialized components, one or more gateways may, in certain embodiments, be configured to serve as a virtual master controller. Moreover, in certain embodiments of the present invention, one or more gateways configured with proxy logic may provide virtual control in both wireline and wireless work machine environments. Fig. 9 is block diagram of an exemplary environment consistent with such embodiments. As Fig. 9 illustrates, gateway 920 may be may be included in a controller-based environment 900. Gateway 920 may be similar to gateway 420 of Fig. 4. In the illustrated system of Fig. 9, gateway 920 may be configured to replace a master controller device that manages the operation of multiple components connected to an on-board network. Gateway 920 may therefore include proxy logic corresponding to various elements included in a master controller. In this exemplary configuration, gateway 920 may interface with different on-board elements (912, 914, 916) through their appropriate data links (e.g., 922). Gateway 920 may also interface with a master controller 955 located off-board and remote to work machine 905, e.g., in a service center 950. A "service center" may include any combination of elements, entities, systems, and controllers for directing, maintaining, and/or servicing various work machines and/or modules implemented in a given environment. Service center 950 and master controller 955 may remotely control operations of a plurality of work

machines and systems, including work machine 905. Gateway 920 may leverage one or more wireless data links to communicate with master controller 955. On-board elements may include any type of modules, components and systems and may be similar to elements 414, 415, and 416 of Fig. 4.

[62] Although three on-board modules or shown in Fig. 9, gateway 920 may be configured to serve as a virtual master controller for any number of modules, both on-board and off-board. Further, although a single work machine and gateway are depicted in Fig. 9, a plurality of geographically dispersed work machines, each including one or more gateways, may be implemented in system 900. In such system configurations, each gateway may serve as an embedded virtual master controller for its respective work machine and may therefore enable a single master controller (e.g., 955) to direct a plurality of geographically dispersed work machines from a central location.

[63] In the environment shown in Fig. 9, gateway 920 may receive and service messages from elements 912, 914, and 916. Consistent with certain embodiments of the present invention, gateway 920 may be configured to intercept messages from data link 922 (e.g., based on source and/or destination address identifiers) and route the intercepted messages to appropriate proxy logic for processing via an address map similar to mapping structure 510. In one embodiment, an intercepted message may be routed to appropriate proxy logic based on its intended destination. Additionally, or as an alternative, intercepted messages may be routed to appropriate proxy logic based on the type of processing needed. For example, messages sent from modules and systems (e.g., 912 , 914, and 916) may all be intended for a master controller but may require different processing. Thus, gateway 920, acting as a virtual controller, may intercept the messages, determine the message types, and route each message, using the address map, to appropriate logic based on the type of processing each message requires. In certain embodiments, gateway 920 may issue commands to modules (e.g., 912, 914) and exchange information with master controller 955 in

service center 950. Gateway 920, may for example, receive messages from master controller 955 and issue corresponding commands to elements 912, 914, and 916. Gateway 920 may also acquire status and diagnostic information from modules and systems (912, 914, 916), e.g., in response to queries from master controller 955. Consistent with embodiments of the present invention, gateway 920 may be pre-configured with proxy logic corresponding to functions associated with master controller 955. Gateway 920 may additionally, or alternatively, dynamically receive proxy logic from master controller 955 or another element within service center 950. In one example, gateway 920 may receive proxy control logic from service center 950 as modules are implemented in work machine 905.

[64] In addition, or as an alternative, to providing proxy control services for on-board and/or off-board components that may be interfaced with a gateway (as discussed above), methods and systems of the present invention may facilitate information exchange among and between proxy logic elements within a given gateway. In both on-board and off-board environments, messages may originate at one or more proxy logic elements located in a gateway and may be destined for another module for which that gateway is serving as a proxy. For example, referring back to Fig. 5, CCM proxy logic 530 may generate (or receive from data link 534) a message intended for an EVIM. Accordingly, gateway 420 may use mapping structure 510 to route the message from proxy logic 530 to proxy logic 540. Such routing may be transparent to both the message source and destination. In addition, referring back to Fig. 7, gateway 715 may transmit a message originating from (or received by) proxy logic included in gateway 715 over data link 722 for receipt by a destination module in work machine 720. Gateway 725 in work machine 720 may serve a proxy for the destination module and may therefore intercept the message and route it to corresponding proxy logic included in gateway 725. Such intercepting and routing may be transparent to the message source and destination.

Protocol Translation

[65] As discussed above, gateway 420 may, in certain embodiments, perform protocol translation processes to facilitate communications between different types of data links, whether on-board or off-board. Gateway 420 may be configured to translate messages received from components interfaced with proxy logic elements into messages consistent with the data link protocol associated with the destination module. For example, using mapping structure 510, gateway 420 may translate message 512 into a message consistent with data link 422's protocol.

[66] As used herein, the term "translating" refers to converting messages from one data link protocol into comparable messages of another protocol. In exemplary translation processes consistent with the present invention, protocol-specific parameters may be translated between different data links. For example, data messages may be translated from an off-board data link protocol (e.g., Ethernet) into data values compatible with an on-board data link protocol (e.g., J1939). Such parameters may include operational parameters, such as engine speed, injection rates, component and/or area temperatures, pressures, etc. corresponding to systems, modules and components located in a work machine environment. Further, parameters may include, or be associated with, engine diagnostic and performance parameters associated with an ECM. Parameters may also reflect commands or be used to perform certain actions. Messages from modules in a work machine environment may include one or more commands to adjust one or more parameter data values based on, for example, a requested action directed to a work machine. In one instance, a message may include a request to increase engine speed of a particular work machine by adjusting, or requesting adjustment of, data values associated with an engine speed parameter.

[67] Consistent with principles of the present invention, a communication application may perform translating processes for any number of

protocols. Messages from multiple and different data links may be discretely or simultaneously translated and sent out on a single data link. Messages may also be received from a single data link and discretely or simultaneously translated and sent out over multiple and different data links. Non-limiting examples of translations include: (1) CDL and J1939 to MODBUS; (2) CDL to ISO11783; (3) CDL to J1939; (4) ATA to J1939; and vice versa. In certain embodiments, gateway 420 may include hardware, firmware, and/or software for performing translation processes. For example, model 300 may include an inter-data link gateway layer 350, including one or more gateway applications 350-1 to 350-T that perform translation processes.

[68] Consistent with principles of the present invention, gateway 420 may maintain a translation data structure, such as a translation table, that maps parameters between data links for facilitating protocol translations. Gateway 420 may access the translation table in order to convert information from one protocol compatible data value to another. In certain embodiments, the translation table may be stored in a memory device within the gateway. See, for example, gateway 120 and its digital core 202's memory 210 accessed by processor 205. Further, the translation table may include a plurality of parameter identifiers (PIDs) representing system parameters associated with various data link protocols. For example, a first PID may represent an engine speed (RPM) parameter associated with certain protocols and a second PID may represent a temperature parameter. The translation table may include any number of different PIDs.

[69] Also, the translation table may also include one or more scaling factors, each representing a data link "view." Each view may correspond to a particular protocol interfaced by gateway 420, such as a proprietary data link view, an Ethernet data link (i.e., Web) view, a J1939 view; and a RS-422 view. The translation table may include any number of views corresponding to data links interfaced by gateway 420. Each "view" may enable its associated data link

to interpret parameter data stored in a universal storage location. This location represents a memory location or locations that store one or more values corresponding to a particular parameter (i.e., parameter data). Parameter data may be received from one or more data links interfaced by gateway 420.

[70] Moreover, each data link view may include a scale factor corresponding to translation logic used by gateway 420 to translate parameter data stored in the US to an appropriate format for the particular data link protocol. In certain embodiments, all views represented by translation table may support a given parameter. For example, an RPM parameter may exist in all of the protocols mapped by the translation table. In some cases, however, certain parameters may be supported by less than all of the views mapped by the translation table. For example, a temperature parameter may be supported by a PDL, Ethernet, and J1939 but not by RS-422. The scale factor for such non-supporting views may be null or set to zero.

[71] In addition, each view in the translation table may include a specific read/write privilege to the universal storage location. That is, certain data links may be assigned write privileges to this location, while other data links have only read access.

[72] Consistent with embodiments of the present invention, the translation table may be pre-configured with a plurality of parameter identifiers and scale factors corresponding to a plurality of data links interfaced by gateway 420. In operation, gateway 420 may receive a message, including a PID and corresponding parameter data, from a particular data link. In response to such a message, gateway 420 may extract the PID and store the parameter data in the universal storage location. In addition, gateway 420 may use the PID to scale the parameter data according to the scale factors included in the translation table, thereby creating multiple "views" of the parameter data. In one example, gateway 420 may receive a request for parameter data from a particular data link. The request may include a PID corresponding to the requested data. In response

to such a request, gateway 420 may extract the PID from the request and scale the requested parameter data (previously stored in the universal storage location) using a scale factor corresponding to the extracted PID and requesting data link protocol.

[73] For additional clarity of explanation, systems 100-400, 700, and 900 are described herein with reference to the discrete functional elements illustrated in Figs. 1-5, 7, and 9. However, it should be understood that the functionality of the illustrated elements and modules may overlap and/or may exist in a fewer or greater number of elements and modules. Elements of each system may, depending on the implementation, lack certain illustrated components and/or contain, or be coupled to, additional or varying components not shown. Moreover, all or part of the functionality of the illustrated elements may co-exist or be distributed among several geographically dispersed locations.

[74] In addition, the steps illustrated in the flowcharts of Figs. 6 and 8 are consistent with exemplary implementations of the present invention. Further, it should be understood that the sequence of events described in Figs. 6 and 8 are exemplary and not intended to be limiting. Thus, other method steps may be used, and even with the methods depicted in Figs. 6 and 8, the particular order of events may vary without departing from the scope of the present invention. Moreover, certain steps may not be present and additional steps may be implemented in Figs. 6 and 8. In addition, it should be understood that the illustrated stages may be modified with departing from the scope of the present invention. For example, in certain embodiments, gateway 420 may be configured to monitor the destination address portion of messages broadcasted over data link 422 in order to determine whether or not a message should be intercepted for proxy processing. Thus, the determination with respect to whether a given message can be locally processed by proxy logic may occur prior to the retrieval of the entire message from the data link.

Industrial Applicability

[75] Consistent with embodiments of the present invention, methods and systems may provide proxy control functions in a work machine environment, whether on-board or off-board. In certain embodiments, a single gateway may be configured as a proxy control module, replacing a plurality of distinct devices. For example, gateway 120 may be configured to replace one or more control modules needed in a particular work machine environment. In such embodiments, gateway 120 may be programmed with proxy logic corresponding to the control modules it replaces. Gateway 120 may be coupled to an on-board data link (which may be a PDL) and leverage a mapping structure to recognize addresses of the control modules for which it is representing as a proxy. Accordingly, a message directed to an address that would otherwise be recognized by an on-board network as associated with an actual control module will be intercepted by gateway 120 and processed accordingly. In this fashion, gateway 120 may serve as a virtual control module.

[76] Systems and methods of the present invention may be leveraged to consolidate features of various special purpose control modules. In certain embodiments, a single gateway may replace a plurality of distinct control modules in a given work machine environment. For example, a single gateway may replace one or more of a CCM, DNTM, EVIM, and a GPSIM. Systems and methods of the present invention may provide services associated with a plurality of specialized components without adapters to multiple on-board networks.

[77] In certain embodiments of the present invention, a single gateway may be configured to serve as a virtual master controller in a given work machine. Such a gateway may include proxy logic corresponding to various functions associated with a master controller. By implementing virtual master controllers in a work machine environment, a single master controller located in a service center may direct a plurality of geographically dispersed work machines via respective gateways embedded in the work machines. In one embodiment, a

master controller may be located in a service center that remotely controls a plurality of work machines and modules. Gateways embedded in the work machines may interact with such a master controller via wireless data links and serve as virtual master controllers for modules and systems in their respective work machines. In this fashion, a plurality of gateways may replace multiple master controllers in a work machine environment.

[78] The embodiments, features, aspects, and principles of the present invention may be implemented in various environments and are not limited to work site environments. For example, a work machine with an embedded gateway may perform the functions described herein in other environments, such as mobile environments between job sites, geographical locations, and settings. Further, the processes disclosed herein are not inherently related to any particular system and may be implemented by a suitable combination of electrical-based components. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.